

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-260714

(43)Date of publication of application : 29.09.1998

(51)Int.Cl.

G05B 19/4093

B25J 19/06

G05B 9/02

G05B 11/32

G05B 19/18

(21)Application number : 09-068280

(71)Applicant : NISSAN MOTOR CO LTD

(22)Date of filing : 21.03.1997

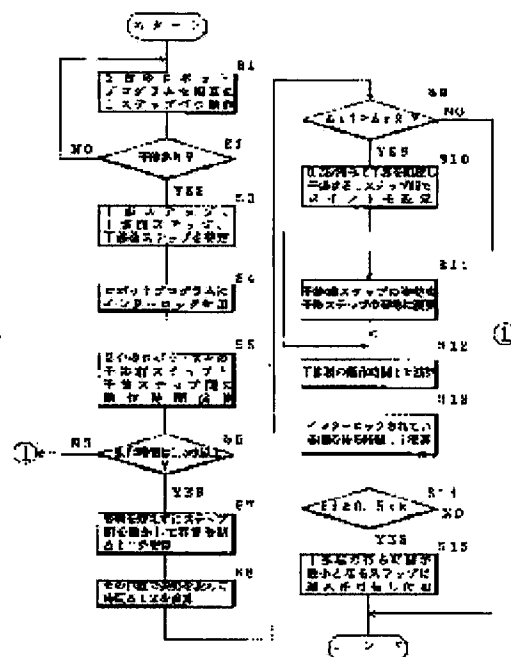
(72)Inventor : SHIMAMURA SHIGEO
IWASA TATSUKI

(54) ROBOT INTERFERENCE AREA SETTING PROGRAM PREPARING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To automatically prepare the program of optimized robot.

SOLUTION: The plural robots are operated while advancing respective programs step by step, the steps of operating area to be mutually interfered with the other robot are found for each robot (S1 and S2) and even when that robot enters the relevant mutual interference operation area, advance permission information permitting the advancement into the relevant mutual interference operation area of the other robot is automatically set to the program of the other robot interfering with that robot (S11-S15).



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-260714

(43) 公開日 平成10年(1998) 9月29日

(51) Int.Cl.⁶

識別記号

F I

G 0 5 B 19/4093

B 2 5 J 19/06

G 0 5 B 9/02

11/32

19/18

G 0 5 B 19/403

B 2 5 J 19/06

G 0 5 B 9/02

11/32

19/18

E

G

A

C

審査請求 未請求 請求項の数 4 O L (全 11 頁)

(21) 出願番号

特願平9-68280

(22) 出願日

平成9年(1997) 3月21日

(71) 出願人 000003997

日産自動車株式会社

神奈川県横浜市神奈川区宝町2番地

(72) 発明者 嶋村 繁生

神奈川県横浜市神奈川区宝町2番地 日産
自動車株式会社内

(72) 発明者 岩佐 達樹

神奈川県横浜市神奈川区宝町2番地 日産
自動車株式会社内

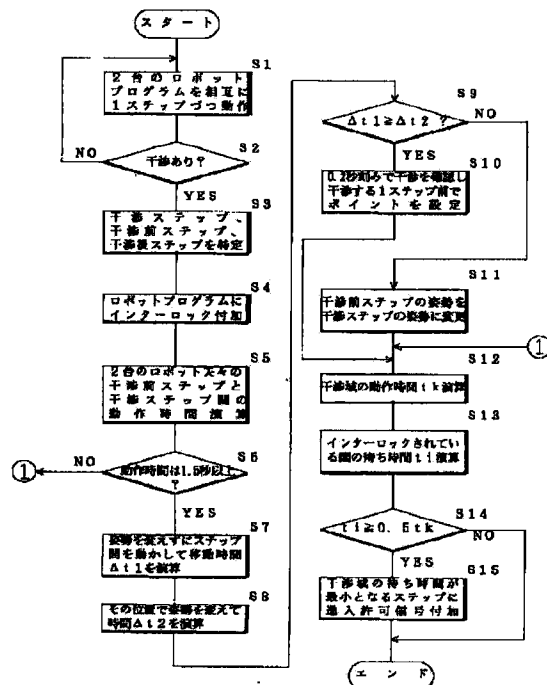
(74) 代理人 弁理士 八田 幹雄 (外1名)

(54) 【発明の名称】 ロボット干渉域設定プログラム作成方法

(57) 【要約】

【課題】 最適化されたロボットのプログラムを自動的に作成する。

【解決手段】 複数のロボットを、それぞれのプログラムを1ステップずつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め (S1、S2)、そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定する (S11～S15)。



【特許請求の範囲】

【請求項1】 互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、

当該複数のロボットを、それぞれのプログラムを1ステップずつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法。

【請求項2】 互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、

当該複数のロボットを、それぞれのプログラムを1ステップずつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定し、

そのロボットが他のロボットと干渉し合う動作領域に入り、第n干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法。

【請求項3】 互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、

当該複数のロボットを、それぞれのプログラムを1ステップずつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定し、

そのロボットが他のロボットと干渉し合う動作領域に入るそのロボットのプログラムの最初のステップに、他のロボットが既に当該干渉し合う動作領域に入っている場合にそのロボットの干渉を回避させるためのインターロック情報を付加し、

そのロボットが他のロボットと干渉し合う動作領域に入り、第n干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに付加し、

また、そのロボットが他のロボットと干渉し合う動作領域から脱出するそのロボットのプログラムの最初のステップに、他のロボットが当該干渉し合う動作領域に入ることを許すインターロック解除情報を付加することを特徴とするロボット干渉域設定プログラム作成方法。

【請求項4】 前記進入許可情報の付加は、この許可情報を任意のロボットのプログラムに付加した場合に、この許可情報が付加されたロボットがこの許可情報が付加された干渉域の動作に要する時間と他のロボットがこのロボットの干渉域の動作のために待たされる時間とを勘案し、他のロボットのその干渉域への進入待ち時間が最短となるようなステップに付加されることを特徴とする請求項2または請求項3に記載のロボット干渉域設定プログラム作成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、互いに干渉し合う動作領域を共有する複数のロボットのプログラムに、他のロボットとの干渉を回避させるための情報を自動的に付加したり、干渉を回避する待ち時間を最小にする情報を自動的に付加したりすることができるロボット干渉域設定プログラム作成方法に関する。

【0002】

【従来の技術】車体スポット溶接工程のように、ワーク周辺に複数台のロボットが密接して設置してある設備の場合には、ロボット相互の干渉を避けるために、ロボット間の動作の優先付けをし、干渉域に後から進入するロボットは、先に干渉域に入った相手のロボットが干渉域を出るまで、干渉域に入る1ステップ前で待機するようにプログラムされる。従来は、このプログラムの作成を、実機を操作して干渉を逐一確認しながら行っている。

【0003】実機を用いて干渉回避のプログラムを作成する手法を2台のロボットの場合について述べる。まず、それぞれのロボットにオペレータが付き、ロボットのプログラムを片方のロボットについて1ステップずつ動作させておおまかな干渉域を把握する。つぎに、両方のロボットを衝突させないように部分的なブレイバックをしながら干渉域のステップを探す。つぎに、どちらのロボットが先に干渉域に入るのかを時間を測定して調べる。最後に、干渉域に入る1ステップ前と1ステップ後に干渉域進入信号および干渉域外信号を設定する。

【0004】

【発明が解決しようとする課題】このような方法で干渉回避のプログラムを作成したのでは、実機の衝突の危険を常に孕んでおり、また、トライアンドエラーによるプログラムの作成となるので、その作成には非常に多くの時間を必要とする。さらに、最適なプログラムの作成（どの位置でどの姿勢で退避させるのか）には熟練を要する。

【0005】また、最近では、グラフィックシミュレータ等を用い、いわゆるオフラインティーチングを行うことによって干渉回避のプログラムを作成することもできるようになったが、干渉域の確認時に実機の衝突のリスクを避けることができるというものの、プログラムの作成が試行錯誤であるということは実機を用いた場合と同じである。

【0006】したがって、干渉をも考慮しなければならないロボットのプログラムの作成には非常に多くの時間を要するという問題と、サイクルタイムを最短とするプログラムの作成を容易に行うことができない（熟練を要する）という問題とがある。

【0007】本発明は、以上のような問題点を解消すべく成されたものであり、互いに干渉し合う動作領域を共有する複数のロボットのプログラムに、他のロボットとの干渉を回避させるための情報を自動的に付加したり、干渉を回避する待ち時間を最小にする情報を自動的に付加したりすることができ、複数のロボットを待ち時間を最小としつつ最小のサイクルタイムで動作させることができるロボット干渉域設定プログラム作成方法を提供することを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するための本発明は、次のように構成される。請求項1に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを1ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、そのロボットが当該干渉し合う動作領域に入っている場合であっても、他のロボットの当該干渉し合う動作領域への進入を許す進入許可情報を、そのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法である。

【0009】このような方法をとれば、他方のロボットが干渉し合う動作領域を出るまで一方のロボットを待たせておかなくとも良くなり、ロボットのサイクルタイムを短縮化することができるようになる。

【0010】請求項2に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを1ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定

し、そのロボットが他のロボットと干渉し合う動作領域に入り、第n干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに自動設定することを特徴とするロボット干渉域設定プログラム作成方法である。

【0011】このような方法をとれば、他方のロボットが干渉し合う動作領域を出るまで一方のロボットを待たせておかなくとも良くなり、またこの領域に入った場合でもロボット同志の干渉を確実に回避しながら、ロボットのサイクルタイムを短縮化することができるようになる。

【0012】請求項3に記載の発明は、互いに干渉し合う動作領域を共有する複数のロボットを画面上で動作させ、それぞれのロボットが干渉せずに動作できるようなプログラムを自動作成するプログラム作成方法であって、当該複数のロボットを、それぞれのプログラムを1ステップづつ歩進させながら動作させ、相互に他のロボットと干渉し合う動作領域がどのステップからどのステップであるのかをそれぞれのロボットについて求め、当該干渉し合う動作領域を分割して複数の干渉域を設定し、そのロボットが他のロボットと干渉し合う動作領域に入るそのロボットのプログラムの最初のステップに、他のロボットが既に当該干渉し合う動作領域に入っている場合にそのロボットの干渉を回避させるためのインターロック情報を付加し、そのロボットが他のロボットと干渉し合う動作領域に入り、第n干渉域内を動作している場合には、他のロボットに対してそのロボットと干渉を起こすことのない干渉域までの進入を許す進入許可情報をそのロボットと干渉する他のロボットのプログラムに付加し、また、そのロボットが他のロボットと干渉し合う動作領域から脱出するそのロボットのプログラムの最初のステップに、他のロボットが当該干渉し合う動作領域に入ることを許すインターロック解除情報を付加することを特徴とするロボット干渉域設定プログラム作成方法である。

【0013】このような方法によれば、複数のロボットを効率的に動作させることができるようになり、結果として、ロボットのサイクルタイムを短縮化でき、作業効率が向上することになる。

【0014】請求項4に記載の発明は、請求項2または請求項3に記載の発明において、前記進入許可情報の付加は、この許可情報を任意のロボットのプログラムに付加した場合に、この許可情報が付加されたロボットがこの許可情報が付加された干渉域の動作に要する時間と他のロボットがこのロボットの干渉域の動作のために待たされる時間とを勘案し、他のロボットのその干渉域への進入待ち時間が最短となるようなステップに付加されることを特徴とするロボット干渉域設定プログラム作成方

法である。

【0015】このように最適なステップに進入許可情報を自動的に付加できるようにすれば、短時間で各ロボットのプログラムを最適化することができるように、プログラムの作成時間を短縮することができるようになる。

【0016】

【発明の効果】本発明は、次のような効果を奏する。請求項1から請求項3の発明では、そのロボットと干渉する他のロボットを、干渉し合う動作領域内であっても、干渉しない範囲内で動作させることができるロボットプログラムを容易に作成することができ、プログラムの作成時間を大幅に短縮することができる。また、この作成はオペレータがいなくても自動的に行われるので、プログラムの作成に携わる人員の削減をすることもできる。

【0017】請求項4の発明では、進入許可情報を付加する最適なステップを自動的に捜し出せるので、熟練を要することなく、だれでも短時間の作業で最適なロボットプログラムの作成が可能となる。

【0018】

【発明の実施の形態】以下に、本発明のロボット干渉域設定プログラム作成方法について説明する。図1は、本発明方法を用いてプログラムされる2台のロボットA、Bを示してある。これらのロボットは、それぞれが図示されているような軌跡を描くようにプログラムされている。このようにプログラムされているロボットを他のロボットとは無関係に独立して動かしたのでは、当然のことながらアーム同志が衝突（干渉するステップは黒丸で、干渉しないステップは白丸で示す）してしまう。これを回避させながら効率良く動かせるようなプログラムを自動的に作成できるようにするのが本発明である。特に、干渉する領域内でも、お互いに干渉しない範囲内でできるだけ双方のロボットが待ち時間なく動作できるようにするプログラムを自動的に作成できるようにしている点が本発明では特筆される。

【0019】本発明の方法を実施する装置は、図2に示すような構成となっている。図中、ロボットプログラム記憶部10は、図1のようにそのロボットの動きのみが書き込まれているプログラムを外部から読み込んで記憶させておくものである。たとえば、ロボットAのプログラムはこの記憶部10のあるエリアに、また、ロボットBのプログラムは記憶部10の他のエリアに記憶される。

【0020】シミュレーション実行部12は、ロボットプログラム記憶部10からそれぞれのロボットA、Bのプログラムを読み込んで、画面上でロボットA、Bを交互に1ステップずつ動かす機能を有するものである。また、各ステップごとの動作時間を計測する機能も持っている。

【0021】干渉チェック部14は、シミュレーション

実行部12によってロボットA、Bを動かした結果、両ロボットA、Bが「干渉を起こすステップ」がどのステップからどのステップであるのかを、それぞれのロボットA、Bについて認識することができるものであり、干渉する前のステップを干渉前ステップとして、干渉するステップを干渉ステップとしてそれぞれ設定したり、サイクルタイムの最適化を図るための新たなステップを作り出したりするための諸機能も有している。

【0022】インターロック自動設定部16は、干渉チェック部14によって認識された「干渉を起こすステップ」に基づいて、両ロボットA、Bのステップにインターロック情報やインターロック解除情報を自動的に付加する指令を出力することができるものである。また、干渉チェック部14における新たなステップの付加指令を受けてそれを出力することもできるものである。さらに、動作時間演算部20の演算結果に基づいて最適なステップに進入許可情報を付加する指令を出力することができるものである。

【0023】ロボットプログラム変換部18は、インターロック自動設定部16からのインターロック情報やインターロック解除情報を付加すべき指令や新たなステップの付加指令、さらには進入許可情報を付加する指令に基づいて、ロボットA、Bの各プログラムにそれらの情報を付加したり、新たなステップを付加したりするために、ロボットプログラム記憶部10に記憶されているそれぞれのロボットA、Bのプログラムをそれらの指令にしたがって変換し、出力することができるものである。

【0024】動作時間演算部20は、ロボットがお互いに干渉し合う動作領域（図1の黒丸の領域）を複数の干渉域に分割し、任意のステップに仮想的に進入許可情報を付加した場合に、それぞれの干渉域の動作にそれぞれのロボットがどの位の時間がかかるのかを演算し、どのステップに進入許可信号を付加した場合がそれぞれのロボットの待ち時間が最小となってサイクルタイムが最小となるかをみつけ出し、そのステップに進入許可信号を付加すべき指令を出力するものである。

【0025】つぎに、本発明にかかるロボット干渉域設定プログラム作成方法の具体的な手順を図3に示したフローチャートに基づいて説明する。まず、ロボットプログラム記憶部10は、図1に示した2台のロボットA、Bのそれぞれのプログラムをそれぞれのエリアに記憶する。そして、シミュレーション実行部12では、この記憶部10に記憶されているそれぞれのロボットのプログラムを1ステップずつ読み出して、2台のロボットを交互に動作させる。この処理によって2台のロボットは画面上で動くことになり、軌跡は例えば図4に示すようになる（S1）。干渉チェック部14では、この2台のロボットが干渉していないかを常に監視し、干渉が検出された場合には、干渉を起こす前のプログラムのステップを干渉前ステップに、干渉を起こしたステップを干渉ス

ステップに、干渉した後のステップを干渉後ステップとする。このステップの特定は、2台のロボットのそれぞれのプログラムに対して行われる。

【0026】たとえば図4に示すように、ロボットAは、ステップaがロボットBとの干渉を起こす直前のステップであるのでこれを干渉前ステップとし、ステップbに進ませるとロボットBとの干渉を起こすので、これを干渉ステップとし、cを干渉後ステップとする。また、ロボットBについては、ステップa'を干渉前ステップに、ステップb'を干渉ステップに、ステップc'を干渉後ステップにそれぞれ特定する。

【0027】なお、これらのステップの特定は、2台のロボットを最初のステップから最後のステップまで動かしてみても行っても良いし、干渉と同時にリアルタイムで行っても良い(S2, S3)。

【0028】以上の処理によって、ロボットAは、そのプログラムのどのステップからどのステップまでの処理においてロボットBと干渉する恐れがあるのか、換言すれば、干渉し合う動作領域がどのステップからどのステップであるのかがわかる。同様にロボットBについてもロボットAと干渉し合う動作領域がどのステップからどのステップであるのかがわかる。

【0029】つぎに、インターロック自動設定部16では、干渉チェック部14によって特定されたそれぞれのロボットのプログラムの干渉前ステップ(a, a')にインターロック情報を付加すべき指令を出し、また、干渉し合う動作領域を脱出した直後のステップ(d, d')にインターロック解除情報を付加すべき指令を出す。これらの情報の付加は、それぞれのステップに対してしても良いし、新たなステップをこれらのステップの前、または後に設けても良い。これらの情報の付加は、インターロック自動設定部16の指令によりロボットプログラム変換部18が行う。すなわち、インターロック自動設定部16からの指令に基づいて、ロボットプログラム記憶部10からそれぞれのロボットA, Bのプログラムを読み込み、それぞれのプログラムの該当するステップにインターロック情報、インターロック解除情報を付加し、これを新ロボットプログラムとして外部に出力する。

【0030】たとえば、ロボットAのプログラムの干渉ステップaに、ロボットBが干渉し合う動作領域に既に入っている場合には、ロボットBがこの動作領域を出るまで、換言すればロボットBのステップd'の処理によってロボットAに対してインターロック解除信号を出力するまで、そのステップで退避させるインターロック情報を付加する。また、ロボットBのプログラムの干渉ステップaに、ロボットAが干渉し合う動作領域に既に入っている場合には、ロボットAがこの動作領域を出るまで、換言すればロボットAのステップdの処理によって

まで、そのステップで退避させるインターロック情報を付加する。さらに、ロボットAのプログラムのステップdには、ロボットBを自由に動作させるためのインターロック解除信号を、ロボットBのプログラムのステップd'には、ロボットBを自由に動作させるためのインターロック解除信号をそれぞれ付加する(S4)。

【0031】さらに、干渉チェック部14では、ロボットA、ロボットBのそれぞれのプログラムに特定された干渉前ステップと干渉ステップとに基づいて、干渉前ステップから干渉ステップに移行するまでに要する時間、すなわち動作時間を演算し、この動作時間が基準時間(この実施形態では1.5秒を想定している)よりも小さければ、ロボットのサイクルタイムのこれ以上の短縮化はできないものと判断して処理を終了する(S5, S6)。

【0032】一方、この動作時間が基準時間よりも大きければ、ロボットの退避位置を干渉し合う動作領域側に移動させることによって、また、姿勢を事前に変化させておくことによってロボットのサイクルタイムの短縮をすることができるようになるので、この場合には次の処理をする。

【0033】まず、干渉チェック部14が干渉前ステップから干渉ステップへの動作時間が1.5秒以上であると判断したときには、シミュレーション実行部12に、ロボットの姿勢(p, q, r)を変えずに干渉前ステップ(x1, y1, z1)から干渉ステップ(x2, y2, z2)までの移動を指示し、干渉チェック部14でこの間の移動時間 Δt_1 を演算する。つまり、ステップ間の平行移動に要する時間を演算する(S7)。

【0034】そしてつぎに、シミュレーション実行部12に、ロボットの位置(x, y, z)を変えずに姿勢のみを変化させる指示をし、干渉チェック部14でロボットの位置を干渉前ステップから動かさずに、ロボットの姿勢を干渉前ステップの姿勢(p1, q1, r1)から干渉ステップの姿勢(p2, q2, r2)に変化させるに要する時間 Δt_2 を演算する(S8)。

【0035】干渉チェック部14では、この演算した Δt_1 と Δt_2 の時間を比較し、 $\Delta t_1 \geq \Delta t_2$ であれば、姿勢を変えるよりもステップ間の移動に時間を要しているのだから移動時間を少しでも短縮することができればロボットの全体のサイクルタイムの短縮化を図ることができるようになるので、干渉前ステップの位置を干渉し合う動作領域側に移動させるようにする。つまり、ロボットが干渉の退避をする位置を干渉ステップ側に移動させる。この退避位置をどこに設けるかは、図5に示すように、時間にして0.2秒分だけ退避位置aを干渉し合う動作領域側(b側)に移動させ(図の軌跡の1刻みに相当)、この点での干渉の確認をし、干渉しなければさらに0.2秒分動かすという操作を繰り返し行い、干渉が確認された点の1つ前の点a1を新たな退避

位置とする。干渉チェック部14は、この新たな退避位置を設けるために、インターロック自動設定部16にこの点に新たなステップを設けるべきことを指令し、ロボットプログラム変換部18はインターロック自動設定部16からのこの指令に基づいて、ロボットプログラム記憶部10から指定のロボットのプログラムを読み込み、そのプログラムの干渉前ステップと干渉ステップとの間に新たなステップa1(図5参照)を挿入する処理をする(S9, S10)。

【0036】一方、 $\Delta t1 < \Delta t2$ であれば、位置を変えるよりもステップ間の姿勢の変化に時間を要しているのであるから姿勢を事前に干渉ステップの姿勢にしておけばロボットの全体のサイクルタイムの短縮化を図ることができるになるので、干渉前ステップの姿勢を干渉ステップの姿勢にするようにする。つまり、ロボットが干渉の退避をしている間に、次のステップ(干渉ステップ)の姿勢に備えておく。

【0037】干渉チェック部14は、干渉前ステップのロボットの姿勢を干渉ステップのロボットの姿勢に変更させるために、インターロック自動設定部16に姿勢変更の指令を出力し、ロボットプログラム変換部18は、インターロック自動設定部16からのこの指令に基づいて、ロボットプログラム記憶部10から指定のロボットのプログラムを読み込み、このプログラムの干渉前ステップの姿勢を干渉ステップの姿勢に変更する処理をする(S11)。

【0038】次に、動作時間演算部20は、ロボットAとロボットBのそれぞれに対して干渉し合う動作領域を動作させるのにかかる時間 t_k を演算する(S12)。また、一方のロボットがこの干渉し合う動作領域に入っている場合に、他方のロボットの待ち時間、すなわちインターロックされている間の動作時間 t_i を演算する。これは、両方のロボットを動かした場合の一方のロボットの待ち時間がどの程度の時間となるのかを調べるためである(S13)。

【0039】この待ち時間 t_i が動作時間 t_k の50%よりも大きい時間であれば、一方のロボットは他方のロボットがこの干渉し合う動作領域から脱出するまでずっと待ち続けなければならないので、サイクルタイムが必然的に大きくなり、ロボットのプログラムとしてはあまり好ましいものではない。したがって、この場合には、ロボットの待ち時間ができるだけ少なくなるように、一方のロボットが干渉し合う動作領域に入っている場合であっても、干渉しないステップまでは他のロボットの動作を許すような進入許可信号を付加する。この進入許可信号を付加する指令は動作時間演算部20が出す。なお、どのステップに進入許可信号を付加するかは最適化の演算は後述する(S14, S15)。

【0040】一方、この待ち時間 t_i が動作時間 t_k の50%よりも小さい時間であれば、許容できる範囲の待

ちの発生であると判断してプログラムの作成を終了する(S14)。

【0041】なお、進入許可信号をプログラム中に付加するか付加しないかの判断は、待ち時間 t_i が動作時間 t_k の50%よりも大きい小さいかによって行っているが、判断の基準となる割合は、ロボットの動作によって任意に設定されるものである。たとえば、一方のロボットが他方のロボットに追従するような形でお互いの干渉域を動作するものであれば、その待ち時間の割合は10%というような少ない値で良いと思うが、一方のロボットが任意の範囲で往復移動、この間他方のロボットを干渉域内に入れることができないような場合には、その待ち時間の割合は70%というような大きな値が選ばれる。

【0042】つぎに、進入許可信号の付加の方法について図6と図7を参照しながら詳細に説明する。この進入許可信号の付加は、次のようなことを可能とするために行われるものである。つまり、追従側のロボットは、図3に示したフローチャートのS1～S11の処理だけでは、優先側のロボットが干渉し合う動作領域の処理を終わるまで待ち状態となってしまふ。たとえば、どちらかのロボットが最初に干渉し合う動作領域に入った場合には、後からこの領域に入ろうとするロボットは、最初に入ったロボットがこの領域から出るまでの間、干渉前ステップでずっと待っていなければならない。確かに、優先するロボットがこの領域を縦横無尽に動き回ってから出るような動きをするものであれば、後のロボットの待ち時間はけっして無駄な時間とはいえないが、優先するロボットが経路を戻ることなく順次進んで行き、後のロボットがこの優先するロボットの軌跡を追いかけて行くような場合には、優先するロボットがこの領域を脱出するまで後のロボットを待たせておくことは非常に無駄である。

【0043】このような場合でも、後のロボットが干渉し合う動作領域内に入ることを許し、優先するロボットに干渉しない範囲内で後のロボットが追従して行けるようにすれば、後のロボットの待ち時間は少なくなるし、作業のサイクルタイムも短縮できる。S11～S15のステップでは、このようなことを可能としているのである。

【0044】さて、図6および図7において、「優先」、「追従」とあるが、「優先」というのは、干渉し合う動作領域に最初に入ったロボットをいい、「追従」というのは、優先するロボットの動作を見ながら後から動作するロボットをいう。この例では、それぞれのロボットの干渉し合う動作領域をA、Bという2つの干渉域に分割している。図6に示すプログラムでは、優先するロボットのプログラムのステップ11とステップ12において、追従するロボットが干渉し合う動作領域(干渉域A+干渉域B)に入れないようにインターロック情報

(SOF4, SOF35)を設定している。したがって、優先するロボットがステップ11、12以降のステップの処理に進んでいる場合には、追従するロボットはこの動作領域に入ることができず、干渉前ステップまでの処理しかできない。すなわち、追従するロボットは干渉する領域に入る直前(ステップ20のSOF4)で待機することになる。

【0045】優先するロボットの処理が進み、ステップ25の処理が終わると、追従するロボットが干渉域Aに入ることが許され(SON, 4)る。このSON, 4の情報10は、干渉域Aへの進入を許可する進入許可情報であるので、追従するロボットはこの情報により干渉域Aに入ることができる。ただ、干渉域Aでの動作が終了しても、干渉域Bには入ることができない。SOF35というインターロック情報がまだ解除されていないからである(ステップ29のSOF35)。

【0046】さらに優先するロボットの処理が進み、ステップ29の処理が終わると、追従するロボットが干渉域Bに入ることが許され(SON, 35)る。このSON, 3515の情報15は、干渉域Bへの進入を許可する進入許可情報である。追従するロボットはこの情報により干渉域Bに入ることができるようになる。ここで、進入許可情報をどのステップに付加するかということがサイクルタイムを短縮する上で非常に重要となる。どのステップに付加するかによって、追従するロボットが待ち時間を少なくして干渉域に入ることができるかどうか左右されるからである。

【0047】この進入許可情報を付加するステップをどこにするかは、次のようにして決める。図7において、追従するロボットが最初のインターロック待ち(ステップ19のPOINT D)でなるべく待ち状態とならないようにするためには、ステップ7~20の処理にかかる時間Tbが優先するロボットのステップ10~25にかかる時間Taよりも大きくなるようにしなければならない。つまり、追従するロボットが干渉域に達する前に優先するロボットがその干渉域を脱出するようにする。ところが時間TbはPOINT Dの位置で決まってしまうので、なるべく優先するロボットのプログラムのステップ24のPOINT Bの設定位置を上の方にしてTaの時間を少なくするようにする。したがって、Tb > Taの範囲内でTBの時間が最大となるようにPOINT Bの位置を決定する。このように設定することができれば、追従するロボットの待ち時間をなくすることができる。

【0048】つぎに、優先するロボットのプログラムのステップ26でPOINT B+1を固定した場合に、追従するロボットが優先するロボットと干渉しないかを、追従するロボットのプログラムのステップ28におけるPOINT Eまでチェックする。つまり、POINT Eよりも前のステップにおいて、優先するロボッ

トのプログラムのPOINT B+1からPOINT Cまでと干渉しないかを確認する。この確認によって追従するロボットのプログラムのステップ19からステップ28までの処理に要する時間Tdが求められる。優先するロボットのプログラムのステップ25からステップ29の処理に要する時間Tcも同様に求められる。

【0049】したがって、2台のロボットを同時に起動させたときのインターロック待ち時間は、次の4通りの式のいずれかによって求めることができる。

① $T_a < T_b$ かつ $T_c < T_d$ の場合には、追従するロボットの待ち時間は0。

② $T_a < T_b$ かつ $T_c > T_d$ の場合には、追従するロボットの待ち時間は $T_c - T_d$ 。

③ $T_a > T_b$ かつ $T_c > T_d$ の場合には、追従するロボットの待ち時間は $T_a - T_b$ 。

④ $T_a > T_b$ かつ $T_c > T_d$ の場合には、追従するロボットの待ち時間は $T_a - T_b + T_c - T_d$ 。

【0050】なお、POINT Bを最初のロジックで仮想的に決定したときに、①の条件を満たす位置(ステップ)ではなかった場合には、この仮に設定したそのPOINT Bの位置を1ステップづつ増やしていった、POINT Cの1ステップ前までについて繰り返し上記の待ち時間を計算し(①~④)、待ち時間を最小とするステップにPOINT B, POINT Eを設定する。

【0051】以上がS15のステップで行われる具体的な内容である。この処理は動作時間演算部20、インターロック自動設定部16、ロボットプログラム変換部18によって分担して行われ、インターロック情報、インターロック解除情報、進入許可情報は最終的にロボットプログラム変換部18によって付加されて、新ロボットプログラムとして外部装置に出力される。

【0052】なお、以上の説明では、2台のロボットが干渉する場合を一例として述べたが、本発明は、これに限られず、複数台のロボットが相互に干渉する場合についても同様に適用可能であることはいうまでもない。

【図面の簡単な説明】

【図1】 本発明方法に使用するロボットの一例を示す図である。

【図2】 本発明方法を実施する装置を示す図である。

【図3】 本発明方法を示すフローチャートである。

【図4】 本発明方法の説明に供する図である。

【図5】 本発明方法の説明に供する図である。

【図6】 本発明方法の特に進入許可信号の付加の説明に供する図である。

【図7】 本発明方法の特に進入許可信号の付加の説明に供する図である。

【符号の説明】

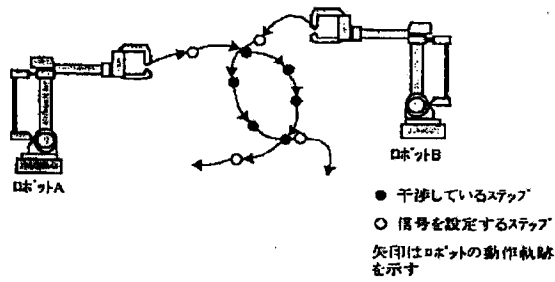
10…ロボットプログラム記憶部、

12…シミュレーション実行部、

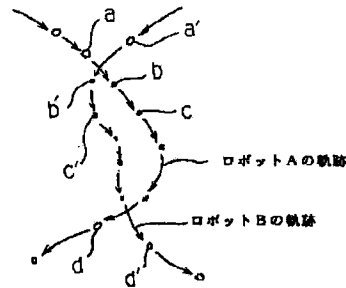
14…干渉チェック部、
16…インターロック自動設定部、

18…ロボットプログラム変換部、
20…動作時間演算部。

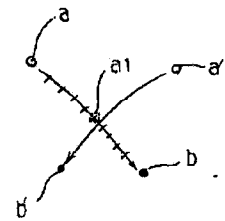
【図1】



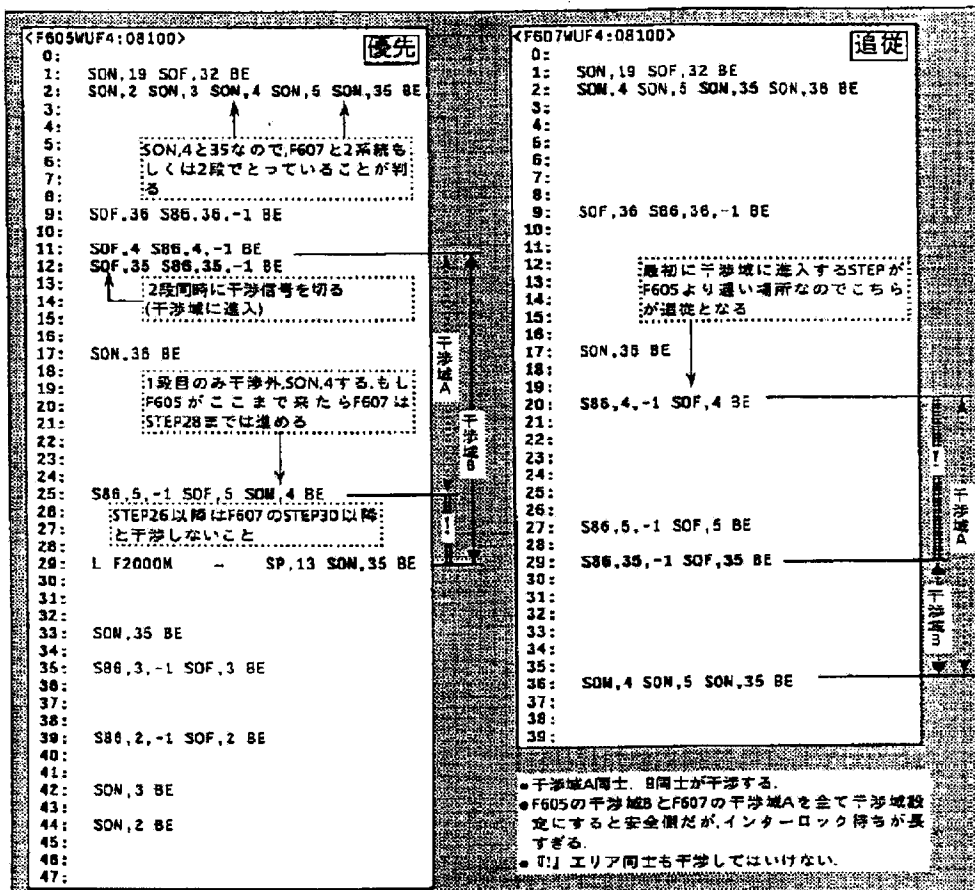
【図4】



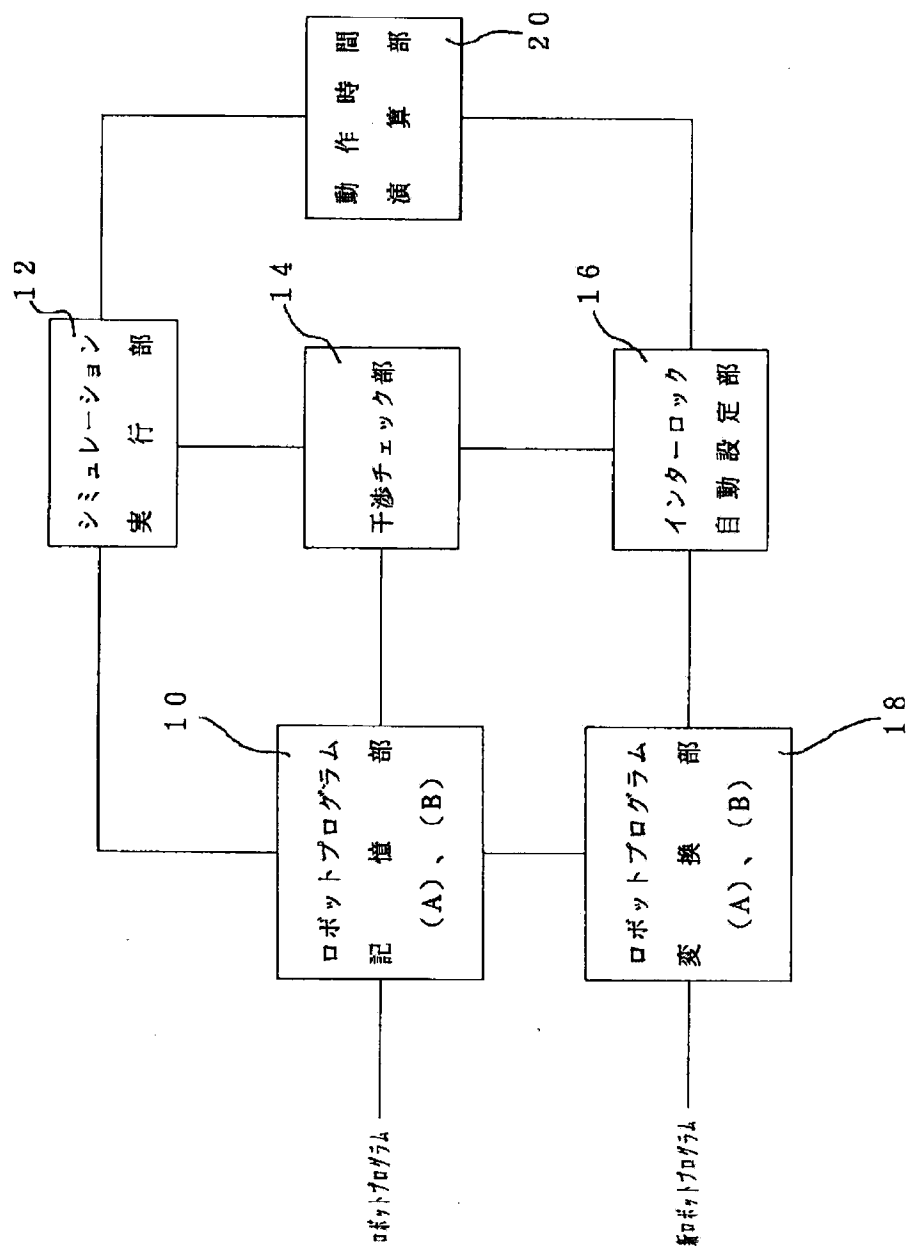
【図5】



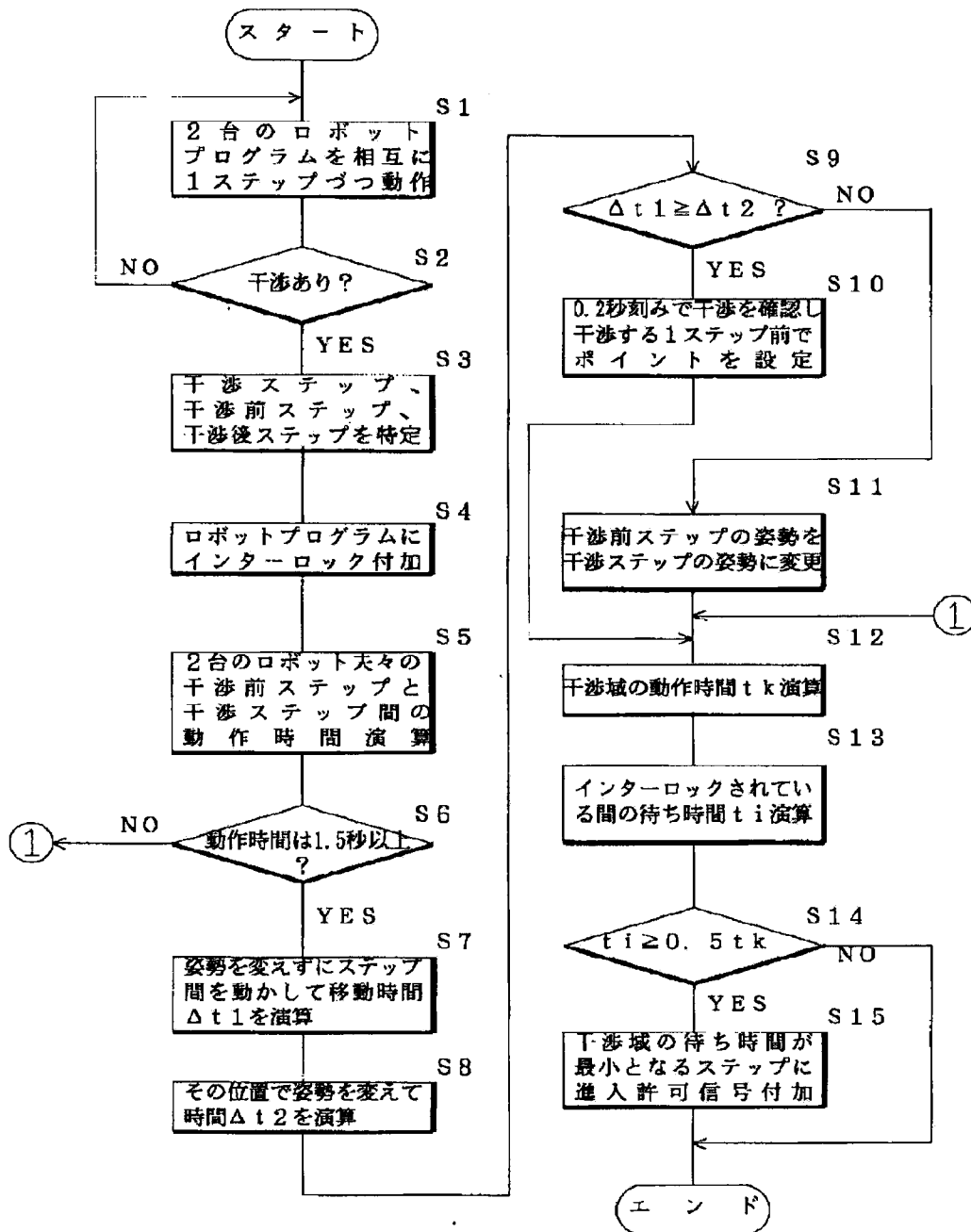
【図6】



【図2】



【図3】



【図7】

